DCOS/UNIX：一个分布式操作系统

华北计算技术研究所　刘京松

摘要　作者提出了一个用于飞行控制环境的分布式操作系统DCOS，讨论了DCOS与通用操作系统的区别，并给出了DCOS的实现方法。DCOS对标准的UNIX操作系统进行了改造和扩充，保持了UNIX与用户的原有界面，在实时调度、时钟管理、实时文件管理、专用设备管理、进程间通讯等方面扩充了UNIX操作系统的功能。

一、引　言

随着微型计算机的迅速发展和广泛使用，出现了分布式计算机系统。

分布式系统是具有多指令流和多数据流的计算机网络。它有如下三个显著特点[5]：

1．系统中的各台计算机没有主次之分，它们高度自治地进行工作。但在工作中又是相互配合，协调工作。

2．系统中采用分散的方式动态地管理和分配多个分布而又相互联系的资源。

3．系统可以并行地运行分布式程序。

分布式计算机系统有着广泛的应用：

1．用于过程控制的分布式系统

过程控制系统主要有工业控制系统和指挥控制系统。例如，Texas Instrument为FAA实现的DABS[16]，此系统为DABS装备的飞机提供了高度可靠的追踪和避免碰撞的功能。这种过程控制系统属于传统的实时控制系统。本文主要讨论用于这方面的一个分布式UNIX[11]操作系统，即DCOS/UNIX (Distributed Cbntrol Operat-ing System)。

2．用于多用户计算的分布式系统

这种系统由多台微机或小型机组成，整个系统为用户提供了一个分时的多用户使用环境，如CM*系统[17]，Roscoe系统[18]，ZCZ系统[7]。

3．用于数据处理的分布式系统

例如银行数据处理系统，飞机订票系统以及办公室自动化等。

在一些较大型的实时控制系统中，特别是飞行控制系统，传统上常常采用大、中型计算机进行集中式控制，或是由多台卫星机和大、中型机组成集中式的控制。但这种系统存在着一些缺陷[2][18]：

1．系统较复杂：一般需要多任务并行处理能力，及时地对大量外围设备进行实时操作。软件复杂，硬件连接非标准且麻烦，这给使用和维护都带来了困难。

2．成本较高

在系统设定后（设定是指系统功能及实现方案的确定），很难在市场出售的产品中找到适中的计算机系统，这就必须对已设定的系统进行专门设计，必然导致研制费用增加，并且研制周期长。

3．可靠性不易解决

中心系统一出故障便影响全局，采用双工冗余设计，代价较高。

4．灵活性差

系统设定后，若在设计上没留余地，则日后很难扩充，而只有更换机器才能增加系统功能。

使用分布式系统代替上述的集中式控制系统，优点如下：

1．可靠性高：LSI器件的使用减少了大量的焊接点，可合理划分系统功能，使单机或几台计算机的故障不致影响全局。易于加入备份系统，提高可靠性。

2．降低成本：这不仅减少了硬件直接成本，也可按需要组成恰如其分的系统，从而减

?1994-2015 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

少了研制费用，缩短了研制周期。

3．系统中多台计算机并行工作：这不仅提高了系统的性能和吞吐量，而且也提高了实时性。例如Intel公司用5台IAPX 432构成的多处理机系统的速度为200万条/秒，其性能超过了VAX-11/780、接近IBM 370/158。

4．易于实现局部冗余技术：这既提高了系统的可靠性，又避免了全局采用冗余控制而增加系统费用。

5．灵活可扩充：随时可按需要加入新的功能，而不致于破坏原有的系统。

另外，分布式系统与集中式系统相比，由于前者中的每台计算机都执行指定的、相对独立的任务，这就导致软件复杂性降低，使得调试简单，提高了软件的可靠性。

但是，分布式系统也有其弱点。对于一些较大的问题，编制一个分布式程序是不容易的，比编制一个顺序程序困难得多。若各机器间信息交流频繁，则不仅增加了系统开销，而且增加了使用的费用，因为通迅费用相当昂贵。

为了实现分布式系统，首先要实现分布式操作系统，我们的作法是对标准的UNIX操作系统进行改造和扩充，构成一个用于控制的分布式控制操作系统，并保持了原来UNIX的功能和界面，称它为DCOS/UNIX。

二、飞行控制系统及实时任务的特点

飞行控制系统在航天飞行控制及国防现代化中发挥着重要作用。例如，1971年，我国成功地发射了第一颗人造卫星，1980年，我国成功地向太平洋预定海域发射了一枚运载火箭，1982年，我国又成功地完成水下发射潜艇导弹试验。

这些飞行控制系统都要求具有快速响应的能力和高度的可靠性，准确无误地监视和控制飞行目标以及地面上的各个子系统的工作情况。它们涉及面广，各子系统间相隔几十米到几百公里，甚至几千公里。

飞行控制系统具有以下特点[2][3][10]：

1．实时性，即时间性，系统能够及时处理和快速响应系统获得的信息。

2．可靠性，要保证系统能够连续运行，但也允许信息的偶然丢失。通常系统中的所有任务都是为一个过程服务的，即使在超载情况下，系统也能完成最重要的任务。

3．简明性，人机会话的简明，甚至一个按钮就是一条命令。

4．运行过程的周期性，即在系统控制下运动的各个任务都是按预先指定的各自周期按时间顺序周期地向前推进的。各个任务的执行不会结束，而是连续不断地为其环境服务，只要该计算机运行的话。

5．有效的纠错能力，对于小错，要进行隔离，必要时要降低功能运行，对于大错，要切换，即切换到备份部件上，继续运行。

6．透明性，在某些方面要保持用户和系统间传输信息的透明性。如用户可对某些外中断信号有直接处理的能力。

7．事件驱动，任务是受一系列可预测的和不可预测事件的驱动。这些事件也许是硬件中断信号，也许是软中断信号。

8．并行性，多个任务是并行执行的，每一任务是为控制某个运动参数而服务的。

UNIX分时系统是在贝尔实验室研究成功的，最初运行在DEC公司的PDP 11系列的小型机上，现在它已被移植到大部分的16位微机上。UNIX简洁、通用、有效、灵活、易于开发，得到了广泛的使用。它还显示了独特的风格[4]：

1．具有可装卸的文件卷的文件系统。整个文件系统形成了从根目录表开始的树形分级结构。用户可把自己的文件系统连入原有的系统上，也可以把它拆卸下来。

2．文件、目录表和外设作为文件统一处理。文件无结构，无类型概念。所有文件均作为无格式的字符流序列，给用户提供了一个简单一致的接口。

3．为用户进程间通讯提供了简单的功能。

4．系统采用模块化结构程序设计，短小

?1994-2015 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

精悍，绝大部分操作系统核心程序都是用C语言[14]书写的。这使用户易读、易改、易扩充、易移植。

5．系统提供了功能完备、使用灵活的命令程序语言，即SHELL语言。

在UNIX得到广泛应用的基础上，贝尔实验室以及国外许多大学、公司在其上开发了新的系统，其中有用于实时控制及开发的MERT实时操作系统[12]，用于网络的网络UNIX[19]，移植到16位微型机上的UNIX变种有Onix、Cnix、Uniflex等，以UNIX为核心的数据库系统INGRES[15]等等。

三、DCOS的特点及它与通用操作系统的区别

由于分布式控制系统的资源分布于各台计算机，因此不能采用通用的单机操作系统的那种集中管理方式，即一类资源仅由一个资源管理来管。

用于控制的分布式操作系统DCOS与通用的操作系统GOS间有着许多相同处，但由于各自面向的用户及环境不同，因而也存在某些差别[1][2][3][9][10][12]：

1．时间驱动：DCOS必须具备按时间周期驱动任务(进程)的功能，而GOS可以省略掉。

2．数据驱动：DCOS始终为用户任务传送最新的实时数据，并驱动用户任务工作，使应用软件始终按最新数据来进行实时处理。GOS一般要保留所有的数据，按顺序分别处理。这里DCOS强调数据的实时性，而GOS却强调数据的顺序性。

3．标准：DCOS没有统一的标准：与用户界面不清，但效率一般比GOS高。因为DCOS简化了许多设计。而GOS主要是为了标准化，以从使用计算机中获得最大效益。

4．结构：DCOS不仅强调作业间的并行，而且组成作业内部的各任务间也必须并行运行，即多任务程度设计(mu ltitasking)。而在GOS中，一作业内的各任务一般是顺序执行的。从结构观点来看，DCOS由于提供了多任务，因而它比GOS更复杂。

5．数据共享：DCOS支持的作业内各任务要求共享一定的公用数据，而GOS不支持此功能。

6．设计目标：DCOS的重点是要在任务转换上取得最低开销，并要求严格的响应时间。而GOS要求能使用户最大限度地利用系统资源，为用户提供广泛的数据处理以及开发工具。DCOS的设计原则不是象GOS那样发挥系统资源的利用率，而是以实时(可靠)为最高设计原则，以取得最低的系统开销。

7．操作：DCOS处于实时操作，严格的时间要求是为了适应专门的硬件。这些响应要求比人机接口的计时需要严格得多。

8．运行：GOS要支持软件开发设施，而DCOS的执行阶段只支持应用软件就可满足要求了，但DCOS在GOS支持的软件开发设施上进行调试和生成。

9．可靠性：DCOS不允许由于某项资源不可靠而使整个系统死锁。为了可靠，通常在DCOS中，用若干台设备替代GOS中的一台。

10．GOS的功能与应用软件分离，而DCOS不完全与应用软件分离，部分功能由用户实现。

为了支持飞行控制系统中的应用软件执行，以提供一个适宜的环境，DCOS必须具备三部分：核心部分，实用程序部分和支持软件部分。

实用程序是一些标准子程序，可供操作系统和用户程序使用。软件支持程序是那些用于开发用户程序的程序，用于修改和维护操作系统的功能。这里主要讨论其核心部分。

核心部分管理分布式系统的各种资源，并控制着所有的用户程序，实用程序及软件支持程序。其功能如下：

1．支持多任务程序设计(在目前的系统中只支持单作业多任务程序设计)，并行执行作业中的多个任务。

2．可根据动态变化的环境，随着时间和实

?1994-2015 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

时数据的 I/O，调度关键性的实时任务。可按时间监督所有的任务处理，确保它们严格的时间周期，使之有节奏地工作。

3．及时接收实时数据，传送给应用软件，并能及时把应用软件处理过的输出数据发送出去。

4．由于系统采用了较坚定的结构，使系统中某几台计算机发生故障时，整个系统仍能工作。

5．允许用户在系统中连接专用的设备，系统具有灵活可扩展的能力。

6．允许动态分布作业。在执行实时任务前，用户作业可不受实际站点数目限制，而根据需要动态分布作业。

7．允许用户进程间的通讯，即提供分布式控制系统站点间信息传输的手段，用以支持资源共享和并行计算。

8．允许系统共享某些文件。

9．允许系统共享某些设备。

四、分布式控制系统模型

在传统的飞行控制系统中，常常采用多台大、中、小型计算机组成一个分布式的计算机网络，其中每台机器作为一个节点，完成予先指定的功能，即这种系统是功能分布系统。在下面的讨论中，我们打算用数台 16 位的同种微型机，如 8086、Z 8000、MC 68000 组成一个个共享总线的局部网，代替上述网中的每一个节点。使用局部网进行分布式控制，有如下优点[6]：

1．任意站点有故障不影响整个分布式系统的工作。数据传送可靠性高，因为没有站间存贮转发手段。

2．成本低：微机的成本比小型机低一个数量级以上。

3．采用总线作为计算机间的通讯手段，灵活易扩充。

在局部的控制网间可根据实时需要连接地理上分散的各子网。局部控制网的拓扑如图 2 所示：

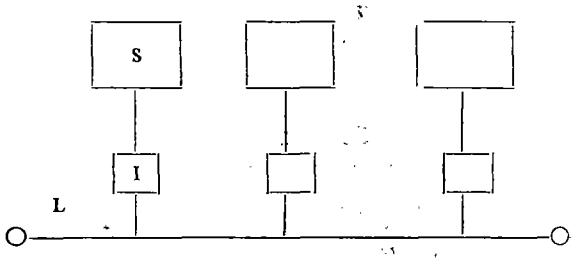


图 1　局部控制网拓扑结构

其中：

S：独立的微型机系统

I：接口处理机系统

L：同轴电缆线

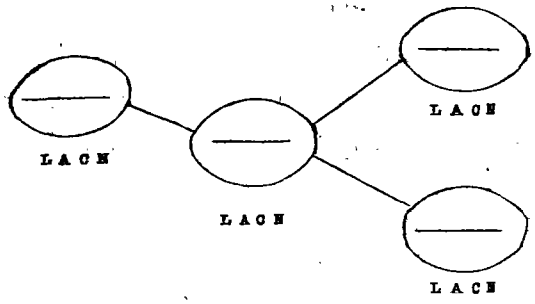


图 2　整个系统布局

其中：

LACN：局部控制网拓扑结构

为了提高系统的可靠性，局部控制网（如图 1 所示）中的各站点均采用双计算机，一台作为主机，另一台作为备分机，在分布式控制中，它们组成双工系统，按主/从方式工作。

系统中所有计算机上的操作系统均采用一致的 DCOS。

系统中的每台计算机都连到一个公用的标准时统站，以使整个系统步调一致地工作。

五、开发 UNIX

虽然 UNIX 操作系统适宜分时环境，具有功能强，使用方便的优点，但并不完全适用于分布式控制环境。为了满足 DCOS 的需要，我们改造并扩充了 UNIX 操作系统核心，这主要

?1994-2015 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

包括：

1．进程调度算法：把 UNIX 中动态的进程调度改为静态进程调度，以满足实时调度的需要。

2．进程控制操作：增加了 9 条系统调用；以扩大用户对其进程的监控能力。

3．时钟管理：增加了一个时钟管理系统，以管理单机系统中的多个物理钟，实现按时间驱动应用软件，使整个系统协调一致地连续工作。

4．文件共享：在 UNIX 文件系统的侧翼增加了一个简单独立的实时文件系统，用它来完成记录实时控制或实验中获取的大量原始数据的功能。用户可利用新增加的 9 条系统调用来对其实时文件进行操作。在此基础上为用户提供了机间共享文件的功能，这是建立在前一部分基础之上的[8]。

5．进程间通讯：分为两部分。在 DCOS 中，建立了两个缓冲池，并提供了申请和释放的操作，以支持进程间的消息传送。在其上，DCOS 实现了信件处理，即完成对用户信件的拆装处理[8]。

6．故障处理：增加故障处理的功能，以保证整个系统在某几个节点出故障时，仍能连续运行。

7．用户设备接口：系统为专用设备定义了统一的数据结构，使 DCOS 易于增添新的设备。

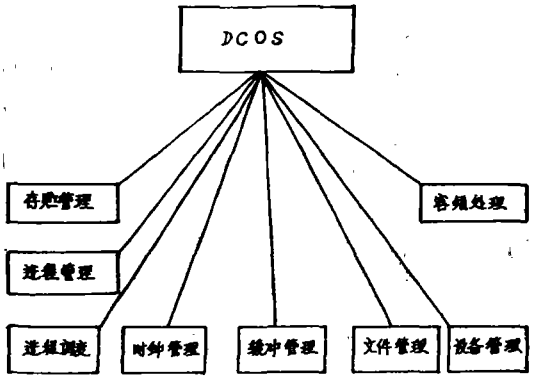
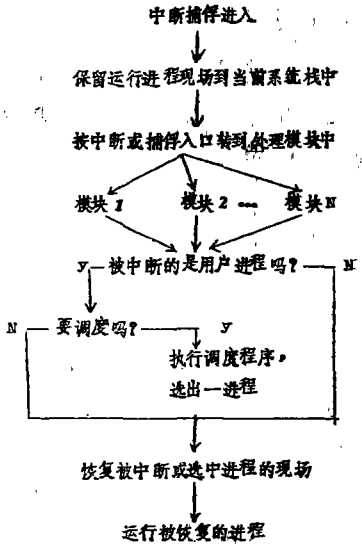


图 3　UNIX 扩充部分示意图

六、进程调度

DCOS 工作流程如下：



DCOS 要求进程调度具备保证实时响应和实时调度关键性进程的功能。在分时系统中，处理机的调度是把处理机给多用户共享，使多个用户进程尽可能平均占用处理机资源，因而进程调度为了权衡多进程公平使用处理机，往往调度开销较大。因而 DCOS 调度和分时调度是有差别的，前者是根据控制对象，有重点的调度，后者是公平对待每一个使用处理机的用户。

在分布式的飞行控制系统中，应用程度是按某种周期重复执行着。这个应用程度就是对反映受控对象运动规律的数学模型的描述，它在整个分布控制过程中，具有被重复执行的特点。用户总是可以将应用程序分成若干个实时任务进程[1][10]，运行于一台或多台机器上。因为数学模型接着某种周期被重事执行，所以这些任务进程也随着数学模型按着某种周期被重复执行。由于所有的任务进程都是为该控制过程服务的，故它们中有些是有联系的，有些是同步的，但它们都是并行执行的，受运动着的控制对象支配。

在所有的任务中，有一个任务最为关键，其它任务都是围绕它而构造的[1]。这些任务

?1994-2015 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

根据其在实时控制中的重要性，可排成一个降级序列 D 1，D 2，……Dn。

DCOS 在进程调度策略上采用按优先级调度，使当前调度到的进程优先级最高，并允许处理机被具有更高优先级的进程抢占。即，只要在就绪的任务进程中出现了一个比当前运行进程优先级更高，它就可以抢占处理机而运行，被抢占的进程处于就绪状态，等待下次得到处理机。

在 DCOS 中，进入进程调度的条件有两个：

1．系统中有进程变化。

2．存在一个就绪进程，其优先级比当前运行进程的优优级高，并设置了重调度标志 runrun，或者有自愿让出处理机的任务进程。

引起进程状态变化的直接原因是中断，可以是内部中断或外部中断。

一个任务进程工作的条件是[1][2]：

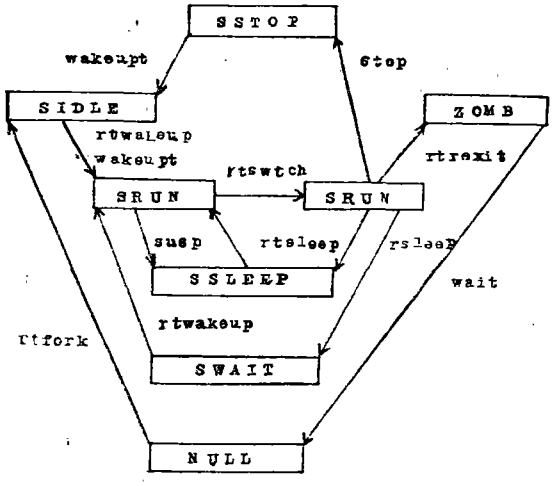
1．时间驱动：把相关于时间的任务进程统一管理，时间一到就去驱动它们运行。

2．数据驱动：外部送来的实时数据积累到一定数量时，就去驱动有关的任务进程去处理。

3．时间或数据取一驱动：对于时间和数据都相关的任务进程，只要有一个条件满足，就去驱动它们工作。这样做，保证了实时信号的及时响应。在时间已到，实时数据尚未到齐时，系统为了及时响应，保证任务进程的周期执行，需要进行实时处理。同样，在数据到齐，时间未到时，系统也需要进行及时处理。

在 DCOS 中，为了满足任务进程按优先级调度的需要，并使调度开销尽可能小，同时对原有的结构改动较小，我们把 proc 表中有定义的项按优先数从小到大进行有序排队。经过排序的 proc 表，使有定义的 proc 项紧凑地排在 proc 前部。因而新的调度程度 rtswtch 只要从 proc 表中顺序扫描出第一个就绪的 proc 项就可以了，即此进程就是当前优先级最高的进程了。排序的功能是由新增加的系统调用 rtfork 完成的，另外，在撤消一进程时，通过执行 rtexit，使 proc 中空闲项被后面的有定义项填上。

下图说明了进程状态间的转换关系：



七、时钟管理

在 DCOS 中，时钟管理的功能如下：

1．调度依赖于时间的进程，能够根据时间对任务进程进行时间驱动或数据驱动。

a．通过建立用户闹钟，控制周期性运行的进程，按其周期把它们投入运行。

b．通过设立定时钟，定时叫醒延迟的任务进程。

c．实现对同步设备管理进程进行驱动，即把与某个同步设备有关的任务进程，在该设备发出帧同步信号时，唤醒它们。

2．维持系统时间，记录各进程的运行时间，提供存取时间的手段。

时间是动态变化、不停地前进的。管理好时间资源，这是 DCOS 的一项重要任务。在 DCOS 中，所有的任务进程都是严格按时间同步的，各进程根据其控制对象所允许的延迟时间限制，具有一定的周期，如有 50 ms 任务进程，100 ms 进程，ls 任务进程等等。DCOS 通过合理管理时间，使各任务进程按其周期被驱动，并行工作，从而实现了按时间调度进程，推动应用软件周而复始的工作。

?1994-2015 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

八、结 束 语

我和阎铭同志经过一年的合作，完成了对DCOS的设计及程序的部分调式工作[8]。

我们是在s/09微型机上模拟调试了约6000行用C语言编制的系统程序，其中包括扩充的实时功能部分，改造的核心部分和进程间通讯部分。但由于条件所限，我们未能进行总体联试。

通过开发UNIX，以实现DCOS。我们认为，UNIX操作系统的模块结构较复杂，希望在今后的工作中，用进程构造DCOS，设计出一个类似于MERT[12]的核心，使DCOS既层次化，功能又独立于局部模块，提高系统的可靠性。

本文是在指导教师刘昌其的指导下写成的，在系统方案的设计中，邵大勋、顾中勋老师提出了很好的建议。在此对他们表示衷心的感谢。

参 考 文 献

[1] 大野豊，"联机实时系统的设计"，1970。
[2] 严友谅，"我国第一颗人造卫星地面指挥控制中心计算机系统"，《计算机工程与应用》，1983年第1期。
[3] W.F.G. Purser, "The Design of a Real-Time Operating System for a Minicomputer", Software-Practice and Experience Vol. 5, 147—167, 1975. SXT-718研制小组，1981年8月。
[4] 刘日升、孙玉方："UNIX操作系统分析报告"，《计算机研究与发展》，Vol. 19, No. 9—10.
[5] 孙钟秀、金兰："分布式计算机系统——计算机研究的一个新方向"，《计算机技术》1983年第1期。
[6] 李性存、王国俊："本地以太网络及其设计"。中国计算机学会首届分布式系统学术讨论会论文集。
[7] "孙钟秀等："一个微型机的分布式系统"，中国计算机学会首届分布式系统学术讨论会论文集。
[8] 阎铭："一种分布式控制系统"。
[9] Andrew St. Johnston "The MACE Technique for Real Time Programming in Small Computers", Software World, Vol. 4, No. 1, MARCH 1973.
[10] S. A. Steel, etal, "Real-Time Tactical Operating System Study". October 1975.
[11] D. M. Ritchie, K. Thompson, "The UNIX Time-Sharing System". Com. ACM Vol. 17, No. 7, July 1974.
[12] BSTL Vol. 57, No. 6, 1978, "MERT: A Multi-Environment Real-Time Operting System".
[13] Per Brinch Hansen, "Distributed Process: A Concurrent Progremming Concept". Com. ACM Vol. 21, No. 10, 1978.
[14] B. W. Kernighan, D. M. Ritchie, "The C Programming Language". Prentice Hall, Englwood Cliffs, N. J., 1978.
[15] ACM SR Vol. 9, No. 5, 1975, "The Design and Implementation of INGRES".
[16] "Distributed Computer Architecture for the Discrete Address Beacon System". October 1—5, 1979.
[17] A. K. Jone, "Software Management of CM*——A Distributed Multiprocessor". AFIPS National Computer Conf. Proc. Vol. 46, 1977.
[18] Solomon M. Finkel, "The Roscoe Distr ibuted Operating System". Proc. of the 7 th Symp. on Operating System Principles, Dec. 1979.
[19] ACM SR Vol. 9, No. 5, 1975, "The Wetwork UNIX System".

?1994-2015 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net